UNCLASSIFIED

AD _**407 791**_

DEFENSE DOCUMENTATION CENTER

FOR

SCIENTIFIC AND TECHNICAL INFORMATION

CAMERON STATION, ALEXANDRIA, VIRGINIA

UNCLASSIFIED

ESD-TDR-63-179

TM-3421

ALGEBRAIC TECHNIQUES OF PATH FINDING AND MINIMUM PATH FINDING IN GRAPHS

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-63-179

May 1963

S Okada   D. Liss   R. Kolker

Prepared for

DIRECTORATE OF SYSTEMS DESIGN

ELECTRONIC SYSTEMS DIVISION

AIR FORCE SYSTEMS COMMAND

UNITED STATES AIR FORCE

L. G. Hanscom Field, Bedford, Massachusetts

Prepared by

ALGEBRAIC TECHNIQUES OF PATH FINDING AND MINIMUM PATH FINDING IN GRAPHS
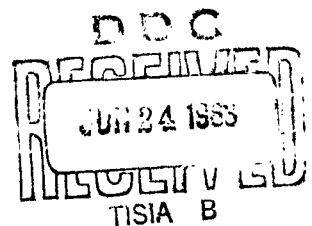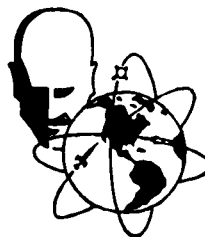
TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-63-179

May 1963

S Okada   D. Liss   R. Kolker

Prepared for

DIRECTORATE OF SYSTEMS DESIGN

ELECTRONIC SYSTEMS DIVISION

AIR FORCE SYSTEMS COMMAND

UNITED STATES AIR FORCE

L. G. Hanscom Field, Bedford, Massachusetts

ABSTRACT

The first part of this paper presents an algebraic
topological approach to finding all paths in a given
graph under various conditions. If a path with any
particular characteristics is desired, such as: mini-
mum distance, least cost, most reliable, etc., it can
be extracted from the list of all paths. In the second
part of the paper, a direct method for finding a mini-
mum path is given. Both techniques are straight-forward
and can easily be performed by a computer.

INTRODUCTION

     This paper treats two problems related to graphs (i.e. diagrams consisting of nodes and branches). The first problem is to find all of the paths in a graph, from any one node to another. The second problem is to find the minimum length path in a graph, from one node to another. The mathematical techniques involved are matrix algebra and topology.

     In the first part of the paper the general path finding problem is handled. At first the theory is presented and then detailed examples are given. In the second part of the paper the minimum path problem is handled. Similarly, the theory is presented at first and then detailed examples are given. Detailed proofs of all theorems are to be found in the appendix.

PART I - GENERAL PATH FINDING

<u>Topological Foundations</u>:

   This part presents a way of finding all paths from one point in a
graph to any other point. This technique is the elaboration of ideas
presented in O-tasc*. Various restrictions can be placed upon the paths
so that they include certain branches and do not include certain other
branches.

    Let us assume a graph to be given and hence its node-branch inci-
dence table. As an example:
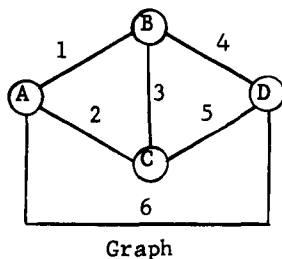
	1	2	3	4	5	6
A	1	1	0	0	0	1
B	1	0	1	1	0	0
C	0	1	1	0	1	0
D	0	0	0	1	1	1

Graph   Incidence Table

<u>Fig. 1</u>

where a one in the ij positions means that node i is incident to branch
j and a zero means that it is not incident. For the details of such
topological concepts see V-AS. The following is a brief summary of the
1-dimensional incidence table and certain of its properties. Let us de-
note a branch or set of branches by its component vector expression.
Thus in the above figure the set of branches 1 and 3 would be denoted by
(101000), where the 1's in the first and third positions mean that
branches 1 and 3 are present and the 0's in the second, fourth, fifth
and sixth positions mean that branches 2,4,5,6 are not present. Similarly
the set of nodes A and D are denoted by (1001). The <u>boundary</u> of a
branch is defined to be the nodes to which this branch is incident. The
boundary of branch 1 is written

$$\partial \ (100000) = (1100) \tag{1}$$

----

* The above abbreviations or acronyms have been found to be a very useful
way of referencing the literature. The acronyms are listed in alphabeti-
cal order in the reference section, with the full reference attached.
The part of the acronym before the hyphen represents the author and the
part after the hyphen the title. Capital letters following the hyphen
signifies a book, and small letters a journal article; if only the first
letter is capitalized, then this signifies an individually available report.

where ∂ is the boundary operator. The co-boundary of a node is defined to be the branches which are incident to this node. The co-boundary of node A is written:

$$\delta \ (1000) = (110001) \tag{2}$$

where δ is the co-boundary operator. Thus the rows and columns of the incidence table correspond to the coboundaries of the nodes and boundaries of the branches.

Let us define an <u>additive</u> operation between two binary component vectors $(a_1,\ldots a_n)+(b_1,\ldots b_n) = (c_1,\ldots c_n)$ in the following manner:

If
    i) $a_i=0$, $b_i=0$ then $c_i=0$

    ii) $a_i=1$, $b_i=0$ then $c_i=1$

    iii) $a_i=0$, $b_i=1$ then $c_i=1$

    iiii) $a_i=1$, $b_i=1$ then $c_i=0$

this can be considered as a modulo 2 position-wise addition.

Let us define the boundary of a set of branches to be the sum of the boundaries of each branch in the set. Let us define the coboundary of a set of nodes to be the sum of the coboundaries of each node in the set.

Let us define a scalar multiplication of vectors by the scalars o and 1 as zero times a vector is the zero vector and 1 times a vector is the vector itself.

It is then quite natural to view the incidence table as a matrix with addition and multiplication between rows and columns as defined above. It can be shown (see V-AS) that these rows form a vector space under the field of integers modulo 2, whose rank is equal to the number of rows minus 1. It can also be shown that the boundary and coboundary operations can be replaced by matrix multiplication involving the incidence matrix, as exemplified in equations 3 and 4.

$$\partial(100000) = H_1 \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 110001 \\ 101100 \\ 011010 \\ 000111 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

$$\delta(1000) = \begin{bmatrix} 1000 \end{bmatrix} \cdot H_1 = \begin{bmatrix} 1000 \end{bmatrix} \cdot \begin{bmatrix} 110001 \\ 101100 \\ 011010 \\ 000111 \end{bmatrix} = \begin{bmatrix} 110001 \end{bmatrix} \tag{4}$$

Let us define two branches b_1, b_2 to be coincident if b_1 and b_2 are incident to the same node at exactly one of their ends.

Let us define a set of branches $b_1, \ldots b_n$ to be a path from a node A to a node B, if b_1 is incident to node A and b_i is coincident with b_{i+1} (i=1,2...n-1), and b_n is incident to node B. Any permutation of the set of branches in a path will also be considered as a path.

Theorem 1: The boundary of a path $b_1, \ldots b_n$ from node A to node B is the set of nodes A and B.

Proof: See Appendix.

Path Calculation

If a graph is given (i.e. its incidence matrix H) and there exists a path p from node A to node B (let us denote the vector expression of these two nodes as n) then

$$H \cdot p = n \qquad (5)$$

Also, any path x from node A to node B must satisfy

$$H \cdot x = n \qquad (6)$$

Thus the general problem of finding a path p from node A to node B is to solve the above matrix equation for x.

The following theory presents a direct method for finding the general solution to equation 6. If the number of rows in H is α then it is easily shown (see V-AS) that the rank of H is $\alpha-1$ and any one of the equations obtained from (6) is linearly dependent on the rest and can be eliminated. Let us call H' the matrix obtained by deleting one of the rows H and n' as the column matrix obtained by deleting the corresponding entry in n. Then (6) is equivalent to:

$$H' \cdot x = n' \qquad (7)$$

Definition: A rectangular matrix of size r x s, where s > r, will be called semi-diagonalized if its left r x r part is a unit matrix and its right r x (s-r) part is an arbitrary matrix of zero's and one's denoted by H_c.
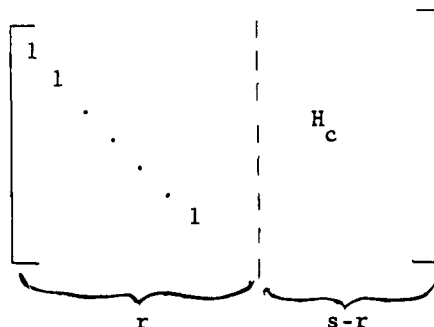
Theorem 2: The matrix H' can be brought into semi-diagonalized form by using the following two operations:

    I   The addition of one row to another

    II The interchange of two columns

Proof: See Appendix

Theorem 3: If H* is the semi-diagonalized form of H' and x* is obtained from x by the corresponding operations of (II) and n* is obtained from n' by the corresponding operations of (I) then the solutions to the matrix equation

$$H* \cdot x* = n* \tag{8}$$

are equivalent to equation (7),

Proof: See Appendix

    The rest of the calculation will be developed by the use of matrices directly.

Let:

$$H* = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \cdot & & & & H_c{}^* \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & & \cdot & \\ & & & & & & 1 \end{bmatrix} \tag{9}$$

$$\underbrace{\hphantom{xxxxx}}_{r} \underbrace{\hphantom{xxx}}_{s-r}$$

$$x* = \left[\!\!\left[x_1{}^* \; \ldots \; x_s{}^* \right]\right.$$

$$n* = \left[\!\!\left[n_1{}^* \; \ldots \; n_r{}^* \right]\right. \tag{10}$$

(The matrix notation $\left[\!\!\left[\; \cdot \; \right]\right.$ means transpose)

From Equation (8)

$$\begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \cdot & & H_c{}^* \\ & & & \cdot \cdot 1 \\ & & & & \end{bmatrix} \begin{bmatrix} x_1{}^* \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_s{}^* \end{bmatrix} = \begin{bmatrix} n_1{}^* \\ \cdot \\ \cdot \\ n_r{}^* \end{bmatrix} \tag{11}$$

which is equivalent to

$$
\begin{bmatrix} 1 & & \\ & \cdot & \\ & & \cdot \\ & & & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1{}^* \\ \cdot \\ \cdot \\ x_r{}^* \end{bmatrix} + \begin{bmatrix} H_c{}^* \end{bmatrix} \cdot \begin{bmatrix} x_{r+1}{}^* \\ \cdot \\ \cdot \\ x_s{}^* \end{bmatrix} = \begin{bmatrix} n_1{}^* \\ \cdot \\ \cdot \\ n_r{}^* \end{bmatrix} \tag{12}
$$

It then follows that

$$
\begin{bmatrix} x_1{}^* \\ \cdot \\ \cdot \\ x_r{}^* \end{bmatrix} = \begin{bmatrix} n_1{}^* \\ \cdot \\ \cdot \\ n_r{}^* \end{bmatrix} + \begin{bmatrix} H_c{}^* \end{bmatrix} \cdot \begin{bmatrix} x_{r+1}{}^* \\ \cdot \\ \cdot \\ x_s{}^* \end{bmatrix} \tag{13}
$$

due to the modulo 2 properties of these matrices. If $(x_{r+1}{}^*, \ldots x_s{}^*)$ is allowed to vary through all binary $(s-r)$-tuples from $(0,0,\ldots0)$ to $(1,1,\ldots1)$ then all solutions for x are obtained. The solution can be brought into a more elegant form by enlarging both sides in the following manner:

$$
\begin{bmatrix} x_1{}^* \\ \cdot \\ \cdot \\ x_r{}^* \\ \hline x_{r+1} \\ \cdot \\ \cdot \\ x_s{}^* \end{bmatrix} = \begin{bmatrix} n_1{}^* \\ \cdot \\ \cdot \\ x_r \\ \hline 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} + \begin{bmatrix} & & H_c{}^* & \\ \hline & 1 & - - - - - - \\ & & \cdot & \\ & & & \cdot \\ & & & & \cdot \\ & & & & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{r+1}{}^* \\ \cdot \\ \cdot \\ x_s{}^* \end{bmatrix} \tag{14}
$$

If both sides are re-ordered by the appropriate interchange of rows and columns on both sides of the equation so as to obtain x in its original order on the left-hand side, then

$$
x = y + H_c \cdot x' \tag{15}
$$

where y is a particular solution for x, and depends only upon the n columns of H' which were used in the semi-diagonalization (which corres-pond to a tree of the graph); H_c depends only on the semi-diagonalized H_c'; x' is a subset of the x and corresponds to the choice of co-tree (complement of a tree, or the $s-r$ columns not in the semi-diagonalized part of $H_c{}^*$).

The total number of solutions for x can easily be obtained. These solutions were found by allowing $(x_{r+1}^* \ldots x_s^*)$ to vary through all s-r tuples of binary numbers. Hence there are $2^{(s-r)}$ solutions for x. If we let n be the number of nodes in the graph, b be the number of branches in the graph, then r = n-1 and s = b. Hence the number of solutions K is:

$$K = 2^{(b-n+1)} \tag{16}$$

Excessive Solutions:

In the previous section we showed how to solve the system of equations in (6). In addition, we showed that any desired path will satisfy (6) and hence be included in the general solutions. It has not been shown that any element of the general solution will be a desired path. This section will serve to clarify this last point.

Let us define a <u>loop</u> as a set of three or more branches $b_1, \ldots b_n$ such that b_i is coincident to $b_{i+1}, i = 1, 2, \ldots n$ and b_n is coincident to b_1, where no subset of the b_i satisfy the same condition. A set of two branches both of whose ends are the same will also be called a loop. Any branches both of whose ends are the same will also be called a loop. Any permutation of a set of branches which form a loop will be considered a loop.

<u>Theorem 4</u>: If b is a set of branches then the equation

$$H \cdot b = 0 \tag{17}$$

is satisfied if and only if b is a loop or a set of loops.

<u>Proof</u>: See Appendix

<u>Definition</u>: If two vectors of zero's and one's

$$a = (a_1, \ldots a_n)$$
$$b = (b_1, \ldots b_n) \tag{18}$$

are given, where a ≠ b and if $a_i = 1$ implies that $b_i = 1$, then a will be said to be included in b, written $a \subset b$.

The following theorem completely cla ifies the solutions to equation (6).

<u>Theorem 5</u>: If b is a set of branches and n is a set of two distinct nodes A and B, then

$$H \cdot b = n \tag{19}$$

if and only if either b is a path from A to B or b is the sum of a path k from A to B and loops $\ell_1, \ldots \ell_r$ where $k \subset b$, $\ell_1 \subset b$, ... $\ell_r \subset b$.

<u>Proof</u>: See Appendix

Thus the solutions to equation (6) are of two kinds; the paths which were required, and the paths plus excessive loops. Once all the solutions have been found, there are several ways of eliminating those solutions which contain excessive loops (excessive solutions).

The first method is based upon the partial ordering induced upon the solutions by the above defined inclusion relation.

<u>Theorem 6</u>: If a set of branches b which is a solution to equation (6) contains loops $\ell_1,\ldots\ell_n$, $\ell_1 \subset b,\ldots\ell_n \subset b$ then there exists a path p such that p is also a solution of equation (10) and $p \subset b$.

<u>Proof</u>: See Appendix

Thus the method for eliminating excessive solutions is as follows: find all of the solutions, then all solutions b such that there exist smaller solutions p (i.e. $p \subset b$) are excessive solutions and should be eliminated.

A second method to eliminate excessive solutions is by rank considerations and can be found in O-tasc 276-9.

<u>Routes Under Specified Conditions</u>:

In general it is possible to specify that certain branches should be included in the path and that certain other branches should not be included. In the original formulation $H \cdot x = n$ the components of x were all assumed to be unknown. Let us now assume that x will be partially specified. Simply set equal to zero those branches which should not be included and set equal to one those branches which should be included. If it is desirable, the general solution of equation (6) can first be found and then these conditions can be used as constraints.

<u>Examples</u>:

Some illustrative examples will help to clarify many of these ideas. Figure 2 is an imaginary telephone communication network. The matrices in Figure 3 are then obtained.

Fig. 2

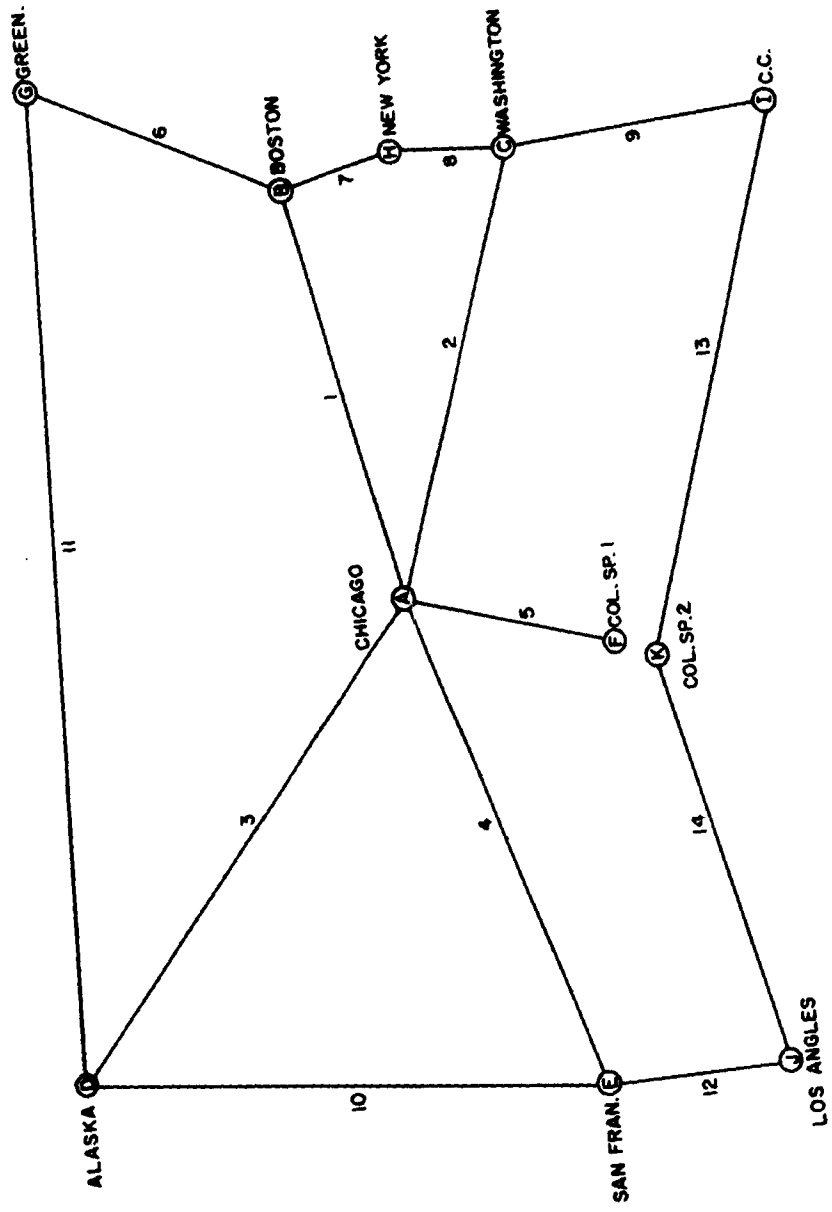IB 10027

Fig. 3

Incidence Matrix H

		1	2	3	4	5	6	7	8	9	10	11	12	13	14
CHICAGO	A	I	I	I	I	I									
BOSTON	B	I					I	I							
WASHINGTON	C		I						I	I					
ALASKA	D			I							I	I			
SAN FRAN.	E				I						I		I		
COL. SP. I	F					I									
GREEN.	G						I					I			
NEW YORK	H							I	I						
C.C.	I									I				I	
LOS ANGLES	J												I		I
COL. SP. 2	K													I	I

INCIDENCE MATRIX H

Reduced Incidence Matrix H' (A-removed)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
B	I					I	I							
C		I						I	I					
D			I							I	I			
E				I						I		I		
F					I									
G						I					I			
H							I	I						
I									I				I	
J												I		I
K													I	I

REDUCED INCIDENCE MATRIX H' (A-REMOVED)

Semi-Diagonalized Matrix H*

	1	2	3	4	5	6	7	9	12	13	8	10	11	14
B+G+H	I										I		I	
C+I		I									I			I
D			I									I	I	
E+J				I							I			I
F					I									
G						I							I	
H							I				I			
I+K								I						I
J									I					I
K										I				I

SEMI-DIAGONALIZED MATRIX H*

Problem 1

Let us find all paths from Alaska to New York. This is the same as finding the general solution to

$$H.x = \begin{bmatrix} 00010001000 \end{bmatrix} \qquad (23)$$

it then follows that

$$H.x^* = \begin{bmatrix} 1010001000 \end{bmatrix} \qquad (24)$$

where

$$x^* = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_9, x_{12}, x_{13}, x_8, x_{10}, x_{11}, x_{14}) \qquad (25)$$

and

$$x = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1010 \\ 1001 \\ 0110 \\ 0101 \\ 0000 \\ 0010 \\ 1000 \\ 1000 \\ 0001 \\ 0100 \\ 0010 \\ 0001 \\ 0001 \\ 0001 \end{bmatrix} . \begin{bmatrix} x_8 \\ x_{10} \\ x_{11} \\ x_{14} \end{bmatrix} \qquad (26)$$

The solutions are easily found by substituting all binary values for $(x_8, x_{10}, x_{11}, x_{14})$:

BRANCHES

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	1		1				1								
2		1	1					1							
3	1			1			1			1					
4		1		1				1		1					
5						1	1				1				
6	1	1					1		1		1				
7			1	1		1	1				1	1			x
8	1	1	1	1		1		1			1	1			x
9	1	1	1	1			1		1			1	1	1	x
10		1	1					1	1			1	1	1	
11	1	1					1		1	1		1	1	1	
12								1	1	1		1	1	1	
13		1		1		1	1		1		1	1	1	1	x
14	1			1		1		1	1		1	1	1	1	
15		1	1			1	1		1	1	1	1	1	1	x
16	1		1			1			1	1	1	1	1	1	x

(SOLUTIONS — row labels at left)

$$(27)$$

The solutions marked with an x mean those with excessive loops. The sixteen diagrams on pages 19 to 22 give the actual sixteen paths.

Problem 2

In this problem let us assume that in Figure 1 Chicago has been destroyed and that we wish to find all paths from Alaska to New York. One way to find the solution would be to start from the beginning and define a new incidence matrix with Chicago removed. However, a much shorter method is possible starting from the solution to Problem 1 in Equation 26. The above condition can be specified by forcing:

$$(x_1,x_2,x_3,x_4,x_5) = (0,0,0,0,0) \qquad (28)$$

in Equation 25. This gives

$$
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
=
\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}
+
\begin{bmatrix} 1010 \\ 1001 \\ 0110 \\ 0101 \\ 0000 \end{bmatrix}
\cdot
\begin{bmatrix} x_8 \\ x_{10} \\ x_{11} \\ x_{14} \end{bmatrix}
\qquad (29)
$$

an independent sub-set of these equations is

$$
\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1010 \\ 1001 \\ 0110 \end{bmatrix} \cdot \begin{bmatrix} x_8 \\ x_{10} \\ x_{11} \\ x_{14} \end{bmatrix}
\tag{30}
$$

and the solution is easily found as

$$
\begin{bmatrix} x_8 \\ x_{10} \\ x_{11} \\ x_{14} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} x_8 \end{bmatrix}
\tag{31}
$$

If this is substituted in Equation 26 we obtain the following two solutions:

1. $P_1 = x_6, x_7, x_{11}$

2. $P_2 = x_8, x_9, x_{10}, x_{12}, x_{13}, x_{14}$

$$\tag{32}$$

Problem 3

Let us find all paths from Alaska to New York which pass x_8 and do not pass x_3. Again we can start from Equation 26. From Equation 26 we obtain

$$
\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0110 \\ 1000 \end{bmatrix} \cdot \begin{bmatrix} x_8 \\ x_{10} \\ x_{11} \\ x_{14} \end{bmatrix}
\tag{33}
$$

The solution is easily found

$$
\begin{bmatrix} x_8 \\ x_{10} \\ x_{11} \\ x_{14} \end{bmatrix} = \begin{bmatrix} 100 \\ 110 \\ 010 \\ 001 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_{11} \\ x_{14} \end{bmatrix}
\tag{34}
$$

If this is substituted in Equation 26 we obtain the following four solutions:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1		1		1				1		1				
2	1	1				1		1			1			
3								1	1	1		1	1	1
4	1			1		1		1	1		1	1	1	1

$$(35)$$

Part II - Determining Minimum Paths

In the study of networks the problem of determining the shortest path (in the network) joining two given points, arises. This problem may be generalized to determining the least cost paths connecting two given points. If a non-negative number s assigned to each branch of a network, then to each path in the network a number called its cost, may be assigned. This number is the sum of the numbers assigned to the branches comprising the path. Various algorithms exist for finding the least cost paths connecting two points of a network ([B-TOG], Ch.7, [L-apa] M-SPTM ,Vol. 2). We shall present a matrix method which is applicable to both directed and undirected networks. In the main body of this paper we shall formulate the method for undirected networks, and in an appendix, the method will be formulated in general.

Let N be a network with a finite number of branches and vertices (nodes). We say that two points of N are joined by a branch if these points are the end points of the branch. We shall assume that no point of N is joined to itself by a branch. By a cost function on N we mean a function f that assigns to each pair of vertices in N, either a non-negative real number or the indeterminate quantity ∞. Furthermore f obeys the conditions

   (1) $f(v,v) = o$ for all vertices v

   (2) $f(v,w) = f(w,v)$ for all vertices v,w

       (This condition will be dropped later).

   (3) If $v \neq w$ then $f(v,w)$ is a non-negative real number if v,w are joined by a branch of N.

   (4) If $v \neq w$ then $f(v,w) = \infty$ if there is no branch joining v to w.
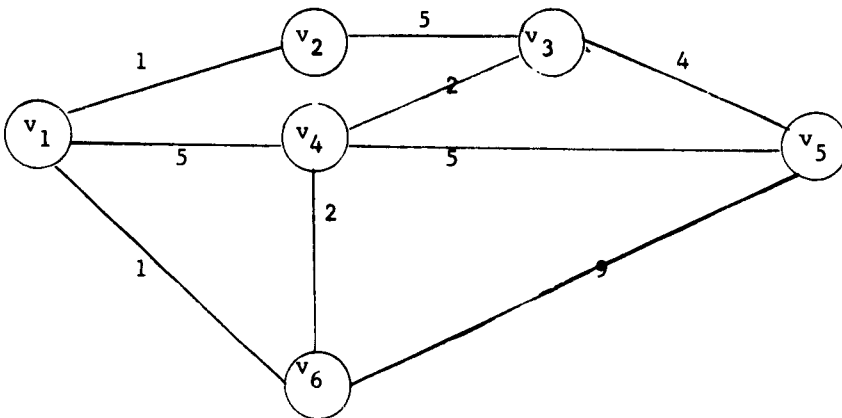
The indeterminate quantity ∞ obeys the conditions:

   (1) $\infty + r = \infty$ for any real number r.

   (2) $\infty + \infty = \infty$

   (3) $\infty > r$ for any real number r.

Let the vertices of N be $v_1, v_2 \ldots v_n$. Then with N we associate the symmetric matrix

$$\begin{bmatrix} f(v_1,v_1) \ . \ . \ . \ . \ f(v_1,v_n) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f(v_n,v_1), \ . \ . \ . \ . \ f(v_n,v_n) \end{bmatrix}$$

Let us consider an example of a network and a cost function on that network.

The matrix associated with this network and cost function is:

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	∞	5	∞	1
v_2	1	0	5	∞	∞	∞
v_3	∞	5	0	2	4	∞
v_4	5	∞	2	0	5	2
v_5	∞	∞	4	5	0	9
v_6	1	∞	∞	2	9	0

This matrix not only tells us what cost is assigned to each branch, but in addition tells us what vertices are joined by branches. For instance, we know v_1 and v_5 are not joined by a branch since the (1,5) entry is ∞.

This follows from conditions 2 and 3 for cost functions.

We shall now define a binary composition on matrices whose entries are real numbers or the indeterminate ∞.

Definition:

$$
\begin{bmatrix}
a_{11} \cdots a_{1n} \\
\vdots \\
a_{n1} \cdots a_{nn}
\end{bmatrix}
*
\begin{bmatrix}
b_1 \cdots b_{1n} \\
\vdots \\
b_{n1} \cdots b_{nn}
\end{bmatrix}
=
\begin{bmatrix}
c_{11} \cdots c_{1n} \\
\vdots \\
c_{n1} \cdots c_{nn}
\end{bmatrix}
$$

where $c_{ij} = \text{Min} \ (a_{i1} + b_{1j}, \ a_{i2} + b_{2j}, \ \ldots \ldots, \ a_{in} + b_{nj})$

Definition: Let A be a matrix whose entries are either real numbers or the indeterminate ∞. Then $A^{(1)} = A$, $A^{(2)} = A * A$, $A^{(3)} = A^{(2)} * A \ldots$ $A^n = A^{(n-1)} * A$.

Lemma 1: Given matrices A, B, C

$$A*(B*C) = (A*B)*C$$

Lemma 2: Given matrix A,
$$A^{(k)} * A^{(k')} = A^{(k+k')}$$

Again let A be the matrix associated with a given network N and let f be a cost function on N. We shall see that the entries of $A^{(n-1)}$, where n is the number of vertices, are the minimum costs of getting from one vertex to another. Consider the matrix $A*A = A^{(2)}$. Let $a_{ij}^{(2)}$ be the i,jth entry in $A^{(2)}$. Then $a_{ij}^{(2)}$ is $\text{Min} \ (a_{i1}+a_{1j}, a_{i2}+a_{2j}, \ldots a_{in}+a_{nj})$ where $a_{ij}=f(v_i,v_j)$. The cost of a least cost path between v_i, v_j consisting of no more than 2 branches is thus given by $a_{ij}^{(2)}$. Similarly the following theorem can be shown.

Theorem 7. Let $a_{ij}^{(k)}$ be the i,jth entry of $A^{(k)}$. Then $a_{ij}^{(k)}$ is the least cost entailed by any path of $\leq k$ branches, joining v_i to v_j.

Now since a least cost path has no loop in it, it suffices to consider the matrix $A^{(n-1)}$ to determine the least cost to get from one point to another. Clearly, any path having n branches or more, must have a loop so our assertion follows from Theorem 7, and condition 3 for a cost function.

Theorem 8: If $k' < k$, then each entry of $A^{(k)}$ is greater than or equal to the corresponding entry of $A^{(k')}$.

Thus for $k>n-1$, $A^{(k)}=A^{(n-1)}$. Let us now consider the network and cost function we gave as an example (See Fig. 1). Let A be their associated matrix. A is a 6×6 matrix so it suffices to consider $A^{(5)}$. But Lemma 2 gives us a simpler way of getting $A^{(5)}$. Notice $A^{(8)}=A^{(4)}*A^{(4)}$, $A^{(4)}=A^{(2)}*A^{(2)}$, and $A^{(2)}=A*A$. By the above remark $A^{(8)}=A^{(5)}$, so we need 3 compositions instead of 4. Here are the compositions done in detail:

$$
\begin{bmatrix}
0 & 1 & \infty & 5 & \infty & 1 \\
1 & 0 & 5 & \infty & \infty & \infty \\
\infty & 5 & 0 & 2 & 4 & \infty \\
5 & \infty & 2 & 0 & 5 & 2 \\
\infty & \infty & 4 & 5 & 0 & 9 \\
1 & \infty & \infty & 2 & 9 & 0
\end{bmatrix}^{(2)}
=
\begin{bmatrix}
0 & 1 & 7 & 3 & 10 & 1 \\
1 & 0 & 5 & 6 & 9 & 2 \\
6 & 5 & 0 & 2 & 4 & 4 \\
3 & 6 & 2 & 0 & 5 & 2 \\
10 & 9 & 4 & 5 & 0 & 7 \\
1 & 2 & 4 & 2 & 7 & 0
\end{bmatrix}
= A^{(2)}
$$

$$
\begin{bmatrix}
0 & 1 & 6 & 3 & 10 & 1 \\
1 & 0 & 5 & 6 & 9 & 2 \\
6 & 5 & 0 & 2 & 4 & 4 \\
3 & 6 & 2 & 0 & 5 & 2 \\
10 & 9 & 4 & 5 & 0 & 7 \\
1 & 2 & 4 & 2 & 7 & 0
\end{bmatrix}^{(2)}
=
\begin{bmatrix}
0 & 1 & 5 & 3 & 8 & 1 \\
1 & 0 & 5 & 4 & 9 & 2 \\
5 & 5 & 0 & 2 & 4 & 4 \\
3 & 4 & 2 & 0 & 5 & 2 \\
8 & 9 & 4 & 5 & 0 & 7 \\
1 & 2 & 4 & 2 & 7 & 0
\end{bmatrix}
= A^{(4)}
$$

$$
\begin{bmatrix}
0 & 1 & 5 & 3 & 8 & 1 \\
1 & 0 & 5 & 4 & 9 & 2 \\
5 & 5 & 0 & 2 & 4 & 4 \\
3 & 4 & 2 & 0 & 5 & 2 \\
8 & 9 & 4 & 5 & 0 & 7 \\
1 & 2 & 4 & 2 & 7 & 0
\end{bmatrix}^{(2)}
=
\begin{bmatrix}
0 & 1 & 5 & 3 & 8 & 1 \\
1 & 0 & 5 & 4 & 9 & 2 \\
5 & 5 & 0 & 2 & 4 & 4 \\
3 & 4 & 2 & 0 & 5 & 2 \\
8 & 9 & 4 & 5 & 0 & 7 \\
1 & 2 & 4 & 2 & 7 & 0
\end{bmatrix}
= A^{(8)}= A^{(5)}
$$

Suppose we wish to find a minimum cost path from v_1 to v_5. The 1,5 entry of $A^{(5)}$ is 8 so there exists a path of cost 8 and this cost is minimum. Now it is clear that if there is a vertex v_j such that the 1,j entry plus the j,5 entry is 8, that v must lie on some minimum cost path. Consider v_4 for instance. The 1,4 entry is 3 and the 4,5 entry is 5. Thus, v_4 lies on a minimum path. That portion of a minimum path connecting v_1 to v_5 which is bounded by v_1 and v_4 must itself be minimum cost. Similarly for the portion between v_4 and v_5. By repeating the above steps we find that $v_1 v_6 v_4 v_5$ is a minimum cost path from v_1 to v_5 and, moreover, it is the only one. In general, though, minimum cost paths are not unique.

If we assign a unit cost to each branch of a network, then the algorithm
just described will yield minimum length paths. This method differs
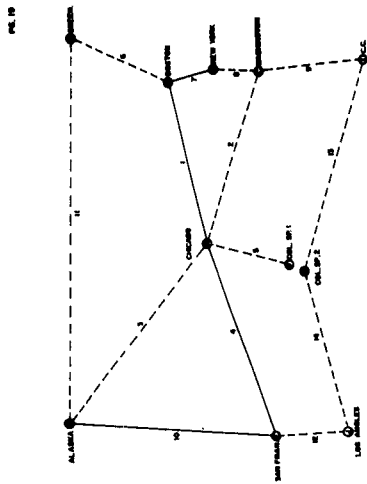from other algorithms in that the two points between which a path is to
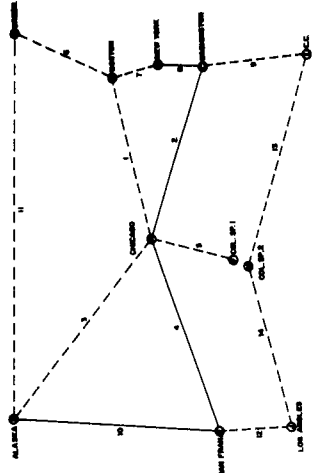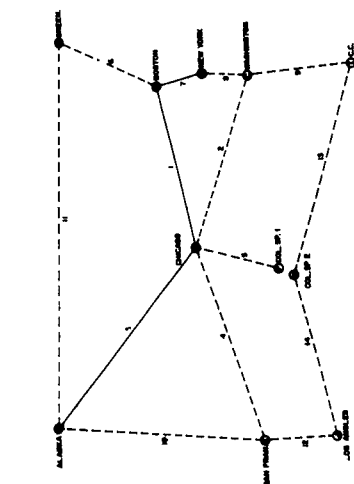be found need not be selected in advance.
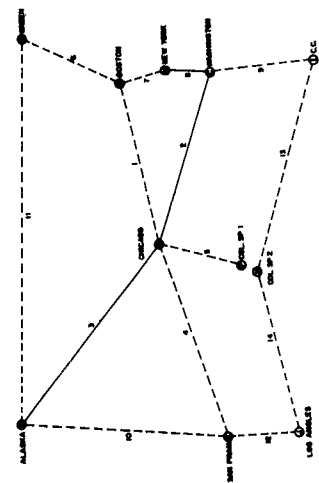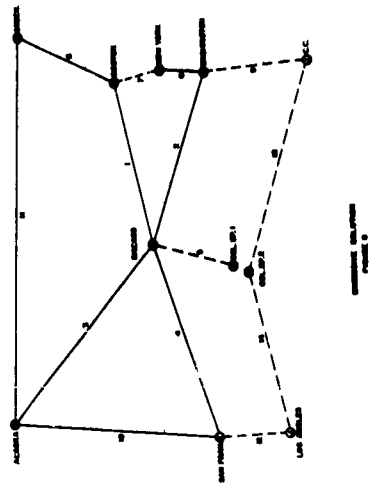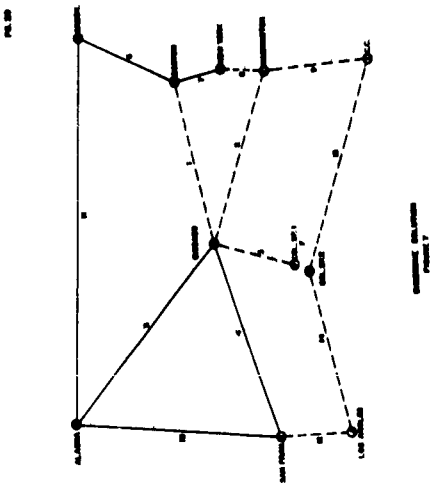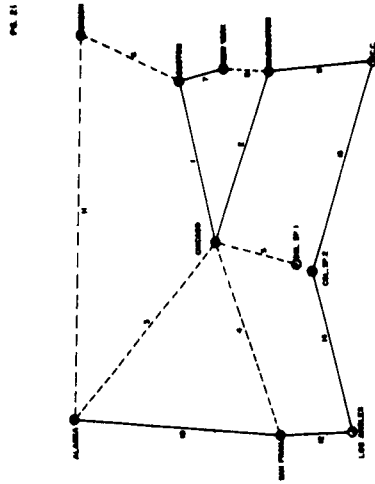
FIGURE 1

FIGURE 2

FIGURE 3
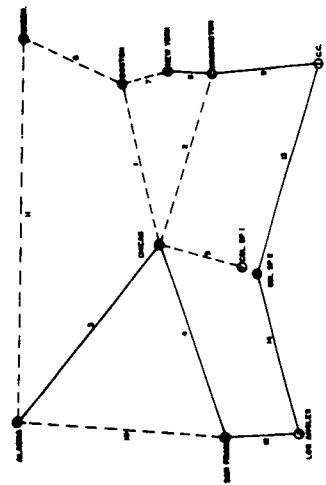
FIGURE 4
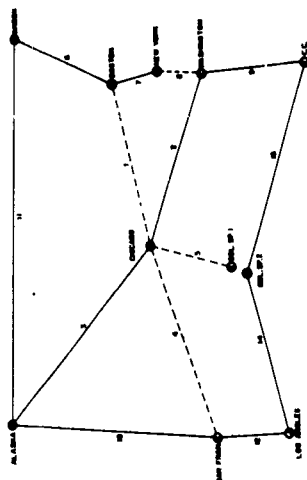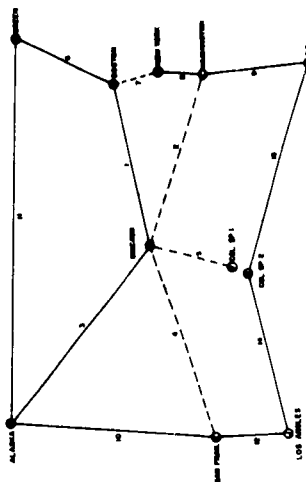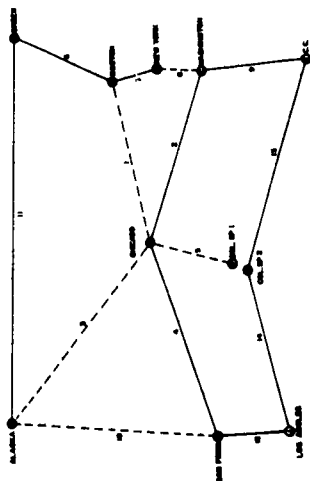
APPENDIX

This appendix contains detailed proofs of all the theorems in the text.

Theorem 1:   The boundary of a path $b_1, \ldots b_n$ from node A to node B is the set of nodes A and B.

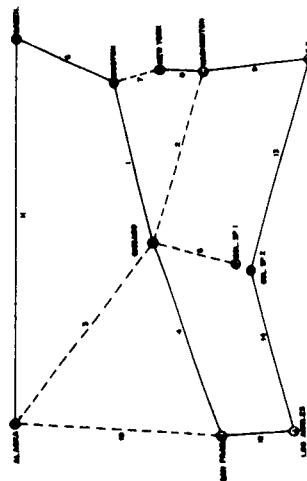Proof: Let us calculate the boundary by adding the boundaries of each branch b_i. Notice that b_i and b_{i+1} (i=1,2,...n-1) are coincident and hence these intermediate nodes cancel out by the modulo 2 addition, leaving nodes A and B as the sum.

Theorem 2:   The matrix H' (an n x m matrix) can be brought into semi-diagonalized form by using the following two operations:

    I.   The addition of one row to another

    II.   The interchange of two columns

Proof: In the first row there is at least one non-zero entry due to the independence of the rows. This can be brought to the 11 position by II. Then the entries 21,31, ... n1 can be made zero by the appropriate addition of the first row to the second, third, ... nth rows. Continuing this with the second row and so on, all non-zero entries below the diagonal are made zero. Then starting with the last row reverse the process and eliminate all non-zero entries above the diagonal.

Theorem 3:   If $H*$ is the semi-diagonalized form of H' and $x*$ is obtained from x by the corresponding operations of II and $n*$ is obtained from n' by the corresponding operations of I, then the solutions to the matrix equation

$$H* \cdot x* = n*$$

are equivalent to those obtained from Equation 7.

Proof: Let us express the matrix Equation 7 in the form of a system of simultaneous equations and analyze the effects of operations I and II on this system:

$$h'_{11}x_1 + \ldots + h'_{1m}x_m = n'_1$$

$$\vdots$$

$$h'_{n1}x_1 + \ldots + h'_{nm}x_m = n'_n$$

The uses of II transform this system to

$$h'_{11}x_1 + \ldots + h'_{1(i+1)}x_{i+1} + h'_{1i}x_i + \ldots + h'_{1m}z_m = n'_1$$

$$\vdots$$

$$h'_{n1}x_1 + \ldots + h'_{n(i+1)}x_{i+1} + h'_{ni}x_i + \ldots + h'_{nm}x_m = n'_n$$

The above is an instance of one application of II, but in general the solutions remain the same. Let us assume an instance of one application of I, where the ith row has been added to the jth row. Then all rows remain the same except for the jth row which becomes:

$$(h'_{i1}+h'_{j1})x_1 + \ldots + (h''_{i\,m}+h''_{j\,m})x_m = n'_i + n'_j$$

Let us show that the solutions to the old system and those of the new system are identical. Assume $(b_1, \ldots b_m)$ to be a solution of the old system, then in the new system all rows other than the jth are obviously satisfied. The jth row becomes

$$(h'_{i1}+h'_{j1})b_1 + \ldots + (h'_{im}+h'_{jm})b_m =$$

$$(h'_{i1}b_1 + \ldots h'_{im}b_m) + (h'_{j1}b_1 + \ldots h'_{jm}b_m) =$$

$$n_i' + n_j'$$

due to the solution satisfying the old system, and hence it satisfies the new system. Now assume that $(b_1, \ldots b_m)$ satisfies the new system, then all rows of the old system other than the jth are obviously satisfied. The ith row and jth row of the new system are

$$h'_{i1}b_1 + \ldots + h'_{im}b_m = n'_i$$

$$(h'_{i1}+h'_{j1})b_1 + \ldots + (h'_{im}+h'_{jm})b_m = n'_i + n'_j$$

Take the sum modulo 2 of these two equations and we get

$$h'_{j1}b_1 + \ldots + h'_{jm}b_m = n'_j$$

and hence the jth row of the old system is also satisfied. This completes the proof.

Theorem 4: If b is a set of branches, then the equation

$$H \cdot b = o$$

is satisfied if and only if b is a loop or a set of loops.

Proof: If b is a loop or a set of loops, then in each loop the boundary operation determines each node (that successive branches are coincident at) twice, hence all these nodes cancel modulo 2 and the boundary is zero.

Assume that a set of branches has a zero boundary. Choose one branch and at one end there must be at least one other branch coincident or the boundary could not be zero. The same holds for the second branch and so on, since this cannot go indefinitely we must finally return to the first branch, hence we have a loop. If this does not exhaust all the branches then repeat the process with one of the remaining branches and at most we have a set of loops.

Theorem 5

If b is a set of branches and n is a set of two distinct nodes A and B then

$$H \cdot b = n$$

if and only if either b is a path from A to B or b is the sum of a path k from A to B and loops $\ell_1, \ldots \ell_r$ where $k \subset b, \ell_1 \subset b, \ldots \ell_r \subset b$.

Proof:

If b is a path from A to B then $H \cdot b = n$. If b is the sum of a path k and loops $\ell_1, \ldots \ell_r$ then

$$b = k + \ell_1 + \ldots + \ell_r$$

and

$$H \cdot b = H \cdot (k + \ell_1 + \ldots + \ell_r) = H \cdot k + H \cdot \ell_1 + \ldots + H \cdot \ell_r$$

$$= n + o + \ldots + o = n$$

Let us assume that $H \cdot b = n$. At least one branch is incident to A, at its other end it is either incident to B or coincident to some other branch (or else this node would be included in the boundary). Continuing this process, some branch will be incident to B. If no more branches remain in b then we have a path from A to B. If some branches remain we choose one of them and it must be coincident to at least one other branch at one of its ends, and so on we find that the remainder constitutes a loop or a set of loops.

Theorem 6

If A set of branches b which is a solution to Equation (6) contains loops $\ell_1, \ldots \ell_n$, where $\ell_1 \subset b, \ldots \ell_n \subset b$, then there exists a path p such that it is also a solution to Equation (6) and $p \subset b$.

Proof:

Assume that a set of branches b which is a solution to Equation (6) contains loops $\ell_1, \ldots \ell_n$, where $\ell_1 \subset b, \ldots \ell_n \subset b$. Choose

$$p = b + \ell_1 + \ldots \ell_n$$

I-3

Then pcb since $\ell_1 \subset b, \ldots \ell_n \subset b$ and also

$$H \cdot p = H \cdot (b + \ell_1 + \ldots + \ell_n) = H \cdot b + H \cdot \ell_1 + \ldots + H \cdot \ell_n$$

$$= H \cdot b + o + \ldots + o = H \cdot b = n$$

hence p is a path.

<u>Lemma 1</u>:

If A, B, C are matrices whose entries are real numbers or the indeterminate ∞ then A*(B*C) = (A*B)*C.

<u>Proof</u>:

Let
$$A = \{a_{ij}\} \quad i,j = 1 \ldots n$$
$$B = \{b_{ij}\} \quad i,j = 1 \ldots n$$
$$C = \{c_{ij}\} \quad i,j = 1 \ldots n$$
$$M = B*C = \{\mu_{ij}\} \quad i,j = 1 \ldots n$$
$$L = (A*B) = \{\lambda_{ij}\} \quad i,j = 1 \ldots n$$

Then $A*(B*C) = A*M = \{a_{ij}\} * \{\mu_{ij}\} = \{r_{ij}\}$

Where $r_{ij} = \underset{k}{Min} \{a_{ik} + \mu_{kj}\}$. $(A*B) \times C = L*C$

$= \{\lambda_{ij}\} * \{c_{ij}\} = \{\bar{r}_{ij}\}$, where $\bar{r}_{ij} = \underset{k}{Min} \{\lambda_{ik} + c_{kj}\}$

But $\mu_{kj} = \underset{k'}{Min} \{b_{kk'} + c_{k'j}\}$, therefore

$r_{ij} = \underset{k}{Min} \{a_{ik} + \mu_{kj}\} = \underset{k}{Min} \{a_{ik} + \underset{k'}{Min} \{b_{kk'} + c_{k'j}\}\}$.

For some k_o, $r_{ij} = a_{ik_o} + \mu_{k_o j}$ and for some k'_o, $\mu_{k_o j} = b_{k_o k'_o} + c_{k'_o j}$.

Therefore, $r_{ij} = a_{ik_o} + b_{k_o k'_o} + c_{i'_o j} = (a_{ik_o} + b_{k_o k'_o}) + c_{k'_o j}$.

However, $(a_{ik_o} + b_{k_o k'_o}) \geq \lambda_{ik'_o}$ and $\lambda_{ik'_o} + c_{k'_o j} \geq \bar{r}_{ij}$ therefore

$r_{ij} \geq \bar{r}_{ij}$. By reversing the proof we can show $\bar{r}_{ij} \geq r_{ij}$ hence $r_{ij} = \bar{r}_{ij}$

which proves the theorem.

<u>Lemma 2</u>:

$$A^{(k)} * A^{(\ell)} = A^{(k+\ell)}.$$

We first show that $A*A^{(k)} = A^{(k)}*A = A^{(k+1)}$. For k = 1, it is true.

I-4

Suppose for $k=k'$, $A*A^{(k')}=A^{(k')}*A=A^{(k'+1)}$, then $A*(A^{(k'+1)}=A*(A^{k'}*A)$ by definition. By Lemma 1, $A*(A^{k'}*A)=(A*A^{k'})*A=A^{(k'+1)}*A$. This induction proves the first assertion. Now $A^{(k)}*A^{(\ell)}=A^{(k)}*(A*A^{(\ell-1)})=A^{k+1}*A^{\ell-1}$. Continuing this rearrangement we get $A^{(k)}*A^{(\ell)}=A^{(k+\ell-1)}*A^{(1)}=A^{(k+\ell)}$.

<div align="right">Q.E.D.</div>

Theorem 7.

Let A be the matrix associated with the network N and the cost function f. Let $a_{ij}^{(k)}$ be the ij entry of $A^{(k)}$. Then $a_{ij}^{(k)}$ is the least cost entailed by any path consisting of $\leq k$ branches joining v_i to v_j.

Proof:

Let $\lambda_{ij}^{(h)}$ be the least cost entailed by any path consisting of $\leq h$ branches and joining v_i to v_j. It is clear that $\lambda_{ij}^{(1)}=a_{ij}$, where a_{ij} is the i,j entry of A, for there is but one path joining v_i to v_j having ≤ 1 branch. Either it is the path having no branches, in which case $\lambda_{ij}^{(1)}=a_{ij}=\infty$ or it is the path consisting of the branch (v_i,v_j) in which case $\lambda_{ij}^{(1)}=f(v_i,v_j)$. Let A be the matrix associated with the network N and the cost function f. Let $a_{ij}^{(k)}$ be the ij entry of $A^{(k)}$ make induction assumption, etc.. Let $\lambda_{ij}^{(k)}$ be the cost entailed by the cheapest path from v_i, to v_j having $\leq k$ branches. Let p be such a path. Either p consists of just the branch (v_i,v_j) or there is a point v_h on p intermediate to v_i and v_j $a_{ij}^{(k)}=\underset{\mu}{Min}\left(a_{i\mu}^{(k-1)}+a_{\mu j}\right)$ by assumption $k>1$. Hence for $\mu=j$ $a_{jj}^{(k-1)}+a_{ij}$ is the cost of a cheapest path from v_i to v_j with $\leq k-1$ branches $\therefore$ $a_{ij}^{(k)}=\lambda_{ij}^{(k)}$. Now suppose for all integers h such that $1\leq h<k$, we have $\lambda_{ij}^{(h)}=a_{ij}^{(h)}$ where $a_{ij}^{(h)}$ is the i,j entry in $A^{(h)}$. Consider $\lambda_{ij}^{(k)}$. There is a path p connecting v_i to v_j and having $\leq k$ branches such that $\lambda_{ij}^{(k)}$ is the cost of p. We assume p has at least one branch. There are two cases to consider:

Case I:

p consists of the branch joining v_i to v_j. Then $\lambda_{ij}^{(k)}=\lambda_{ij}^{(k-1)}=\lambda_{ij}^{(k-1)}+o=\lambda_{ij}^{(k-1)}+a_{jj}=a_{ij}^{(k-1)}+a_{jj}\geq Min(a_{ih}^{(k-1)}+a_{hj})=a_{ij}^{(k)}$. If for some $h\neq j$, we have $\lambda_{ij}^{(k)}>a_{ih}^{(k-1)}+a_{hj}$ then we may conclude there is a path of $\leq k-1$ links going from v_i to v_h and then going to v_j by

<div align="center">I-5</div>

one more branch whose cost is less than $\lambda_{ij}^{(k)}$. This contradicts the minimality of the cost of p.

Case 2:

There is a third point v_ℓ intermediate to v_i and v_j through which p passes. Let t be the number of branches on that part of p connecting v_i to v_ℓ. Then we see that $\lambda_{ij}^{(k)} = \lambda_{i\ell}^{(t)} + \lambda^{(k-t)} = a_{i\ell}^{(t)} + a_{\ell j}^{(k-t)} \geq a_{ij}^{(k)}$. Again as on Case 1, if $\lambda_{ij}^{(k)} > a_{ij}^{(k)}$ there exists h such that $\lambda_{ij}^{(k)} > a_{ih}^{(t)} + a_{hj}^{(k-t)} = \lambda_{ih}^{(t)} + a_{hj}^{(k-t)} \geq \lambda_{ij}^{(k)}$.

Q.E.D.

Theorem 3:

If $k \geq k'$ then the i,j entry of $A^{(k)} \leq$ the i,j entry of $A^{(k')}$ for all i,j.

Proof:

The i,j entry of $A^{(k')}$ is $\lambda_{ij}^{(k')}$ by the previous theorem. If we consider longer paths (consisting of more branches) connecting v_i, v_j, we may come up with a cheaper path, in which case $\lambda_{ij}^{(k)} < \lambda_{kj}^{(k')}$ or we may not, in which case $\lambda_{ij}^{(k)} = \lambda_{ij}^{(k')}$.

Q.E.D.

In view of lemma 2 and theorem 2, if we are dealing with a network of n nodes, the most matrix compositions we need to do, in order to determine all minimum costs between nodes, is $\log_2 n + 1$. The procedure for getting $A^{(n-1)}$ is to perform the "doublings" $A^{(2)}, A^{(4)}, \ldots A^{(2^{\log_2 n+1})} = A^n = A^{(n-1)}$.

Once we have $A^{(n-1)}$ we can easily determine a minimum cost path connecting v_i, v_j. If $A_{ij}^{(n-1)} = \infty$ there are no paths connecting v_i, v_j. If $a_{ij}^{(n-1)} \neq \infty$ then look for a node v_{h_1} such that $h_1 \neq j$, $h_1 \neq i$ $a_{h_1 j}^{(n-1)} \leq a_{hj}^{(n-1)}$ for all $h \neq j$ and for which $a_{ih_1}^{(n-1)} + a_{h_1 j}^{(n-1)} = a_{ij}^{(n-1)}$. If no such node exists then the branch connecting v_i to v_j is already a minimum cost path. If v_{h_1} does exist then there is a branch of the network N connecting v_{h_1} to v_j. In that case we repeat the procedure to find a minimum cost path connecting v_i to v_{h_1}, etc..

Illustrations:

I-6

D. M. Liss
D. M. Liss

S. Okada
S. Okada

Robert J. Kolker
R. J. Kolker

REFERENCES

1. B-TOG: C. Berge:"Theory of Graphs," J. Wiley & Sons, New York, 1962.

2. L-apa: C.Y. Lee: "An Algorithm for Path Connections and Its Appli-
   cations", IRE Trans. PG.EC-10.3 September 1961, 346-65.

3. M-SPTM: E. F. Moore: "The Shortest Path Through a Maze", Proc.
   Inter. Symp. on The Theory of Switching",Harvard Univ. Press, 1957.

4. O-tasc: S. Okada: "Topology Applied to Switching Circuits", Proc.
   Symp. on Information Networks, P.I.B., Brooklyn, pp. 267-290, April,
   1954.

5. V-AS: O. Veblen: "Analysis Situs", Amer. Math. Soc. Coll. Pub.,
   Vol. V, Part 2, Amer. Math. Soc., N.Y., 1931.

Hq. ESD, L.G. Hanscom Field, Bedford, Mass.
Rpt. No. ESD-TDR-63-179. ALGEBRAIC TECHNIQUES OF PATH FINDING AND MINI-MUM PATH FINDING IN GRAPHS. Final report, May 1963, 31p. incl. illus., 5 refs.
Unclassified Report

The first part of this paper presents an algebraic topological approach to finding all paths in a given graph under various conditions. If a path

1. Matrix algebra
2. Topology
3. Mathematical computer data
I. Project No. 707
II. Contract AF33(600)-39852
III. The MITRE Corporation, Bedford, Mass.
IV. Okada, S.
V. Liss, D.

Hq. ESD, L.G. Hanscom Field, Bedford, Mass.
Rpt. No. ESD-TDR-63-179. ALGEBRAIC TECHNIQUES OF PATH FINDING AND MINI-MUM PATH FINDING IN GRAPHS. Final report, May 1963, 31p. incl. illus., 5 refs.
Unclassified Report

The first part of this paper presents an algebraic topological approach to finding all paths in a given graph under various conditions. If a path

1. Matrix algebra
2. Topology
3. Mathematical computer data
I. Project No. 707
II. Contract AF33(600)-39852
III. The MITRE Corporation, Bedford, Mass.
IV. Okada, S.
V. Liss, D.

Hq. ESD, L.G. Hanscom Field, Bedford, Mass.
Rpt. No. ESD-TDR-63-179. ALGEBRAIC TECHNIQUES OF PATH FINDING AND MINI-MUM PATH FINDING IN GRAPHS. Final report, May 1963, 31p. incl. illus., 5 refs.
Unclassified Report

The first part of this paper presents an algebraic topological approach to finding all paths in a given graph under various conditions. If a path

1. Matrix algebra
2. Topology
3. Mathematical computer data
I. Project No. 707
II. Contract AF33(600)-39852
III. The MITRE Corporation, Bedford, Mass.
IV. Okada, S.
V. Liss, D.

Hq. ESD, L.G. Hanscom Field, Bedford, Mass.
Rpt. No. ESD-TDR-63-179. ALGEBRAIC TECHNIQUES OF PATH FINDING AND MINI-MUM PATH FINDING IN GRAPHS. Final report, May 1963, 31p. incl. illus., 5 refs.
Unclassified Report

The first part of this paper presents an algebraic topological approach to finding all paths in a given graph under various conditions. If a path

1. Matrix algebra
2. Topology
3. Mathematical computer data
I. Project No. 707
II. Contract AF33(600)-39852
III. The MITRE Corporation, Bedford, Mass.
IV. Okada, S.
V. Liss, D.

with any particular characteristics is VI. Kolker, R.
desired, such as: minimum distance,   VII. In ASTIA
least cost, most reliable, etc., it   collection
can be extracted from the list of all
paths. In the second part of the
paper, a direct method for finding a
minimum path is given. Both tech-
niques are straight-forward and can
easily be performed by a computer.

---

with any particular characteristics is VI. Kolker, R.
desired, such as: minimum distance,   VII. In ASTIA
least cost, most reliable, etc., it   collection
can be extracted from the list of all
paths. In the second part of the
paper, a direct method for finding a
minimum path is given. Both tech-
niques are straight-forward and can
easily be performed by a computer.

---

with any particular characteristics is VI. Kolker, R.
desired, such as: minimum distance,   VII. In ASTIA
least cost, most reliable, etc., it   collection
can be extracted from the list of all
paths. In the second part of the
paper, a direct method for finding a
minimum path is given. Both tech-
niques are straight-forward and can
easily be performed by a computer.

---

with any particular characteristics is VI. Kolker, R.
desired, such as: minimum distance,   VII. In ASTIA
least cost, most reliable, etc., it   collection
can be extracted from the list of all
paths. In the second part of the
paper, a direct method for finding a
minimum path is given. Both tech-
niques are straight-forward and can
easily be performed by a computer.